Intellectual Property Rights Management using a Semantic Web Information System

Roberto García, Rosa Gil, Jaime Delgado

Universitat Pompeu Fabra (UPF), Departament de Tecnologia,
Pg. Circumval·lació 8, E-08003 Barcelona, Spain
{jaime.delgado, roberto.garcia, rosa.gil}@upf.edu

Abstract. IPR (Intellectual Property Rights) Management is a complex domain. The IPR field is structured by evolving regulations, practises, business models,… Therefore, DRMS (Digital Rights Management Systems) are very difficult to develop and maintain.
The NewMARS DRMS is our contribution to this field. A knowledge oriented approach has been chosen in order to make this development capable of dealing with this complicated domain. This requirement and the objective of easy Web integration have made the Semantic Web technologies the best choice.
NewMARS is a semantics enabled metadata managing system. Metadata is associated to IPs (Intellectual Properties) using URIs and it is structured using web ontologies. There are descriptive, rights and e-commerce ontologies for the different views on IPs. Semantic enabled metadata is then used to facilitate content providers to publish intellectual properties offers and customers to find and automatically negotiate purchase conditions.
All NewMARS modules are interrelated using the ontologies shared semantics. This has allowed developing very flexible project infrastructures that facilitates easy adaptation to new IP e-commerce scenarios.

1 Introduction

This research tries to make a contribution to the Intellectual Property Rights (IPR) Management field. IPR Management has been strongly affected by the digital era changes. Even now, all the new situations related to the Intellectual Property arisen from digitalisation and the Internet have not been satisfactorily resolved yet.

Some of these problems are faced by current initiatives trying to solve interoperability between Digital Rights Management (DRM) systems. They are systems responsible for managing digital rights like digital content IPR. DRM systems started from isolated and proprietary initiatives. However, they are lately moving to a web-broad application domain due to the World Wide Web effect on the digital content market.

One of the main initiatives is MPEG-21[1], a MPEG standardisation framework for digital contents management. MPEG-21's IPR modelling part is divided into the Rights Expression Language (REL) [2] and the Rights Data Dictionary (RDD) [3].

There are many other initiatives but, basically, all have one thing in common, they work at the syntactic level. Their approach is to make a formalisation of some XML DTDs and Schemas [4] that define a rights expression language (REL). In some cases, the semantics of these languages, the meaning of the expressions, are also provided but formalised separately as rights data dictionaries (RDD). Rights dictionaries list terms definitions in natural language, intended for human consumption and not easily automatable.

However, this kind of syntactic approaches are not solving the problem as a whole. They do not scale well in really wide and open domains like the Internet. Therefore, the interoperability problems are reappearing as it is very difficult to establish one "fit all requirements" standard. For instance, the OMA (Organisation for Mobile Applications) has tried a REL different from MPEG-21 one. For OMA the choice is ODRL that has been proposed to W3C [5].

Most probably, we are not going to see a clear winner in the REL battlefield, at least in the short time range. However, automatic processing means for the huge and heterogeneous amounts of metadata produced by DRM are required. The syntax is not enough when unforeseen expressions are met. Here is where machine understandable semantics come to help metadata interpretation to achieve interoperability.

Our idea is to facilitate the automation and interoperability of IPR frameworks integrating both parts, the Rights Expression Language and the Rights Data Dictionary. These objectives can be accomplished using ontologies, which provide the required definitions for the right expression language terms in a machine-readable form. Thus, from the automatic processing point of view, a more complete vision of the application domain is available and more sophisticated processing can be carried out.

We have taken the Semantic Web approach [6] because it is naturally prepared for the Internet domain and thus we use web ontologies [7]. The modularity of web ontologies allows their easy extension and adaptation to meet evolvability and interoperability.

Once we chose the Semantic Web approach we proceeded to develop an IPR Ontology, IPROnto [8,9,10]. However, the ontology is only a formalisation without utility if it is not put into practice. This has been the objective of the NewMARS project: to build an IPR Management utility that takes profit from the advantages of the IPROnto formalisation, which will facilitate the implementation of digital content e-commerce solutions.

2 Application Domain

In order to effectively put NewMARS into practice, what has been done first is to analyse the IPR business model. This business model defines the environment where NewMARS will fit, the actors with which it will interact and the interaction rules. The business model we have considered is presented in subsection 2.1.

The NewMARS Project planning has been guided by the idea to make a knowledge guided development, from a computer point of view. This implies transferring a great amount of the development effort from the functional model to the domain knowledge model.

Consequently, the number of application functions is reduced to some basic ones in charge of message interchange among the application parts. A user actions diagram detailing actors and functions is detailed in subsection 2.2. Therefore, the focus is placed on the semantics of these messages.

As it has been introduced before, IPROnto is used as the basis of the knowledge model. Therefore, a great part of this effort has been already done and it is reused in NewMARS. There are only some small extensions to the knowledge model derived from the practical aspects of the project. More details about this are given in subsection 2.3.

2.1 Business Model

The e-commerce of IPR is guided by a business model that has emerged from the associated regulations framework, the commercial activity and the electronic means that have influenced it.

In order to build NewMARS upon a quite generic and flexible business model, the one defined as a result of the detailed work carried out in the IMPRIMATUR Project [11] has been the foundation. The NewMARS Business Model identifies a set of basic roles and interactions among them. These basic roles of the IPR activity are shown in Fig. 1. They constitute the value chain.

In parallel, some support services have been also identified. They constitute the basic services that facilitate the IPR e-commerce activity. They are shown in Fig. 1 around the roles to which they give support along the whole value chain.

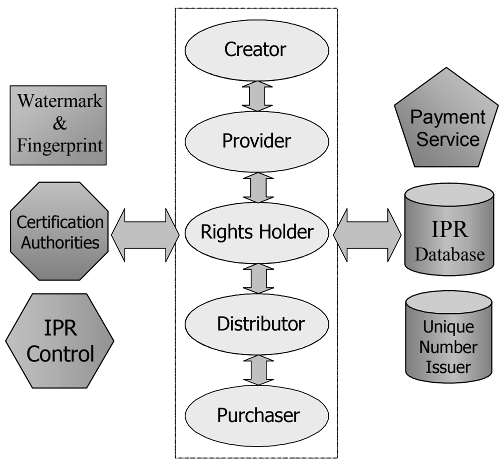

Fig. 1. Generic IPR Business Model

To facilitate the implementation of this model, it has been combined with a broker-based e-commerce model that has been extensively tested in previous research [12,13,14]. The final broker-based business model implemented in NewMARS is shown in Fig. 2.

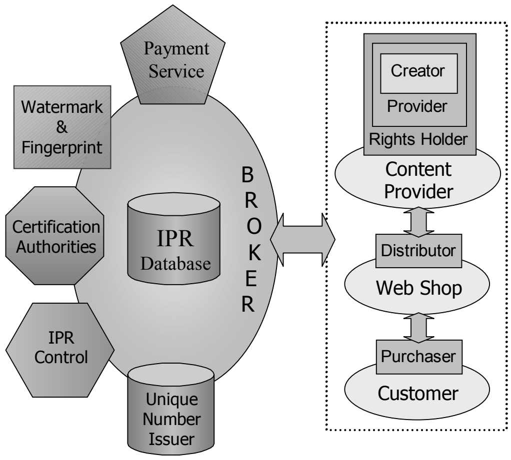

Fig. 2. NewMARS Broker-based IPR Business Model

The broker facilitates value chain actors access to the IPR e-commerce services. Moreover, in the NewMARS scenario, actors have been simplified to three, each one playing one or more roles: Content Provider (it can play Creator, Provider and Rights Holder roles), Web Shop (it plays Distributor role) and Customer (it plays Purchaser role).

In addition to the broker, the NewMARS project is also going to implement the Distributor role through a web shop. Consequently, there will be only two external actors: Content Provider and Customer. More details are given in the user actions analysis in the next section.

2.2 User Actions Analysis

Fig. 3 shows the actions that specify the relations among the external actors that have been identified and the application.

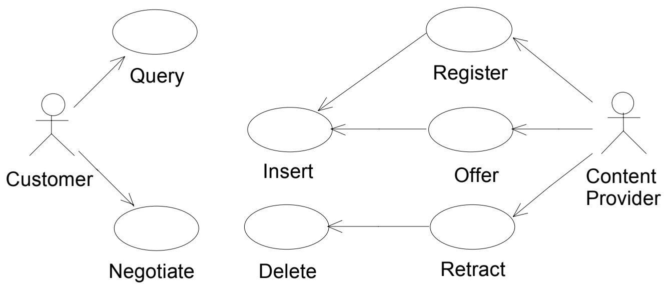

Fig. 3. User actions diagram

The user actions are detailed below:

- **Insert**: this is an "internal" action that is not directly accessible to external actors. Its functionality is accessed from other actions. Basically, what this action does is to store information about a resource into the NewMARS system. Due to the knowledge-oriented approach this action can be viewed as the assertion making one.
- **Delete**: it is also "internal" but it is the counterpart of the previous action. It is responsible for un-asserting facts.
- **Register**: content providers use this to add new information about the intellectual properties (IPs) they manage. The information chunks are sets of assertions describing the IPs and their rights situation.
- **Offer**: this action is accessed by the Content Provider to add e-commerce information about an IP.
- **Retract**: content providers can delete information chunks about IPs they have previously inserted in NewMARS. This includes descriptive, rights and e-commerce information.
- **Query**: customers can use this action to look for desired IPs. The queries submitted by the customer are matched against descriptive, rights and e-commerce information stored in NewMARS. In return, the customer receives all the registries associated to the resources that have matched the criteria.
- **Negotiate**: once e-commerce information has been retrieved, if it does not completely satisfy the customer it can be negotiated. When a satisfactory offer is achieved the customer can accept it, then it is fulfilled.

2.3 Metadata

The IPs information that is managed by NewMARS is modelled as metadata associated to resources. There is also a set of ontologies that provide the required semantics. As it has been introduced before, IPROnto is used as the foundation for rights and e-commerce metadata.

However, descriptive metadata depends on the particular IPs that are managed. Due to project requirements, NewMARS was planned considering digital multimedia content. Therefore, ontologies about descriptive metadata for this kind of content where considered.

The MPEG-7 standard [15] was taken as the source for the descriptive ontology due to its coverage and relevance. First of all, an RDF Schema modelling MPEG-7 characterisation of multimedia content types [16] was reused. However, it only covered a small part of MPEG-7. Then, it was complemented with a 350 genres ontology generated automatically from some MPEG-7 multimedia description schemes [17].

The previous descriptive ontologies provide a quite satisfactory framework for multimedia content description. Finally, the multimedia specific aspects are complemented with the generic ones provided by Dublin Core [18]. An example of RDF metadata description is shown in Table 1.

Table 1. IP metadata example including descriptive, rights and e-commerce information


```xml
<!DOCTYPE rdf:RDF […
    <!ENTITY mp7 'http://metadata.net/harmony/MPEG7/MPEG7.rdfs#'>
    <!ENTITY mp7g http://dmag.upf.edu/ontologies/2003/03/MPEG7Genres.rdfs#'>
    <!ENTITY ipr 'http://dmag.upf.edu/ontologies/2003/12/ipronto.owl#'>
    <!ENTITY cur 'http://www.daml.ecs.soton.ac.uk/ont/currency.daml#'>
    <!ENTITY nms 'http://dmag.upf.edu/ontologies/2003/05/NewMARS.rdfs#'>
]>
<rdf:RDF xmlns:rdf="&rdf;" xmlns:rdfs="&rdfs;" xmlns:dc="&dc;" xmlns:mp7="&mp7;"
xmlns:mp7g="&mp7g;" xmlns:ipr="&ipr;" xmlns:cur="&cur;" xmlns:nms="&nms;">
<mp7:Video rdf:about="urn:newmars:30m-USAP">
    <rdf:type rdf:resource="&mp7g;Documentary"/>
    <dc:title xml:lang="ca">També més que un club</dc:title>
    <dc:description xml:lang="es">Seguimiento de…</dc:description>
    <dc:language>ca</dc:language>
    <dc:date rdf:datatype="&xsd;date">1999-05-16</dc:date>
    <dc:format>video/mpeg</dc:format>
    <dc:creator><rdf:Bag>
        <rdf:li>Guardia, Carles</rdf:li><rdf:li>Pou, Francesc</rdf:li>
    </rdf:Bag></dc:creator>
    <dc:publisher rdf:resource="http://www.tvcatalunya.com"/>
</mp7:Video>
<ipr:Offer rdf:about="http://dmag.upf.es/newmars/offer19990611-103520">
    <ipr:offerer>NewMARSSeller@dmag.upf.es:1099/JADE</ipr:offerer>
    <ipr:time rdf:datatype="&xsd;date">1999-06-11</ipr:time>
    <ipr:patient>
        <ipr:PurchaseLicense>
            <ipr:licenser rdf:resource="http://www.tvcatalunya.com"/>
            <ipr:permission><ipr:Access>
                <ipr:place rdf:resource="http://www.tvcatalunya.com/online/30m-USAP"/>
                <ipr:patient rdf:resource="urn:newmars:30m-USAP"/>
                <ipr:timeFrom rdf:datatype="&xsd;date">1999-07-01</ipr:timeFrom>
                <ipr:timeTo rdf:datatype="&xsd;date">2004-07-01</ipr:timeTo>
            </ipr:Access></ipr:permission>
            <ipr:obligation><ipr:Compensation>
                <ipr:payee rdf:resource="http://www.tvcatalunya.com"/>
                <ipr:input>
                    <ipr:CurrencyMeasure rdf:value="1">
                        <nms:currencyUnit rdf:resource="&cur;EUR"/>
                    </ipr:CurrencyMeasure>
                </ipr:input>
            </ipr:Compensation></ipr:obligation>
        </ipr:PurchaseLicense></ipr:patient>
</ipr:Offer></rdf:RDF>
```


Another key element about metadata in NewMARS is that it is expected to come from many different sources, i.e. metadata stores. Therefore, it is required that the metadata management processes implemented support this feature. However, from the outside, the users should experience an integral view of metadata so the metadata must be merged transparently.

In order to implement this feature, the best option is to use RDF metadata through all the NewMARS information flows. Therefore, NewMARS receives RDF metadata as input, manages it and also produces RDF metadata as output. When RDF metadata coming from different sources must be combined, the RDF graph model facilitates metadata integration that is reduced to a process of graph merging. Once integrated, the metadata graph can be serialized and sent to the output. More details about how this is implemented are shown in section 3.1.3.

3 Development

Once the application domain has been introduced, this section details how the application has been developed. The driving force has been knowledge orientation. This has been materialised by prioritising application modules decoupling and basing module interrelation in shared semantics.

Web technologies, and more concretely Semantic Web tools, have been chosen as the more appropriate ones considering these requirements. First of all the following technological choices have been realised:

- Message transport: SOAP [19].
- Message encoding: RDF [20].
- Message semantics: ACL [21].
- Ontology language: OWL [22].
- User interface: HTML.
- Negotiation: JADE+JESS [23,24].

From the combination of requirements, design principles and technological choices, the architecture shown in Fig. 4 has emerged.

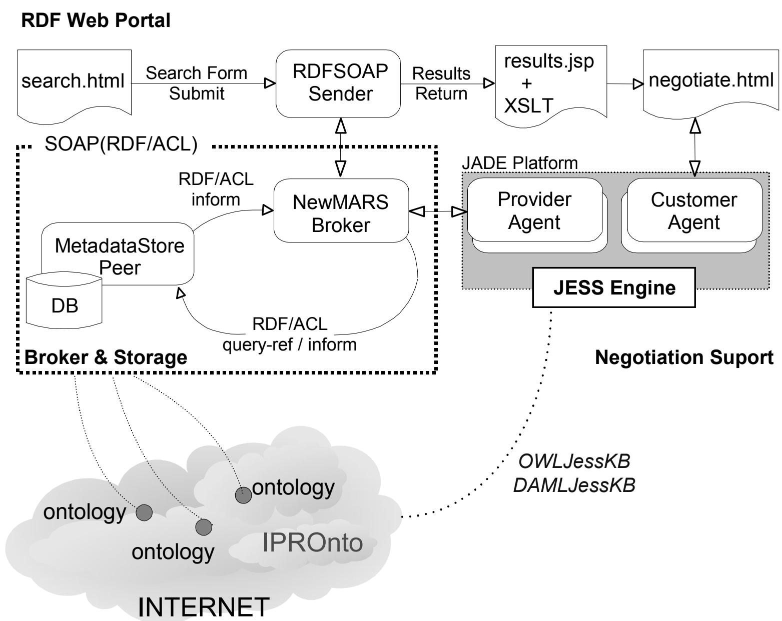

Fig. 4. NewMARS architecture

The architecture defines three main blocks:

1. **Broker and Storage**: this block is in charge of the main NewMARS responsibilities, i.e. all actions apart from “Negotiate”. The broker offers a SOAP interface through which it interchanges SOAP messages. These messages are

encoded using RDF and then structured using a web ontology that models FIPA ACL (Agent Communication Language) in order to provide message semantics. Message semantics define which messages are queries, facts assertions or facts removals. In each case, independent metadata stores are accessed for metadata retrieval, insertion or deletion. More details in section 3.1.

2. **Web Portal**: this block is the front-end that interacts with external users. The objective of this block is to provide an easy and common user interface, so HTML has been selected. In order to interact with the broker the RDFSOAPSender has been developed. First, RDFSOAPSender facilitates sending messages to the broker: it encapsulates HTML forms submissions as RDF/ACL messages and sends them using SOAP to the broker. Second, it manages messages responses: it processes the return messages in order to transform their RDF content to HTML that can be shown to the user. This block is detailed in section 3.2.
3. **Negotiation Support**: this block is responsible of giving service to the "Negotiate" action. The objective is to offer automatic or semiautomatic negotiation support to users. Agents' technology is used to perform this. We have chosen JADE as the multi-agent platform because it provides agent technology building-blocks and implements FIPA standards. Agents' decision support is managed by the JESS expert system. More details are given in section 0.

3.1 Broker and Storage

As it has been introduced before, the broker block of NewMARS has a SOAP interface. However this interface is only used for message transport. Thus, message semantics do not depend on different SOAP interface methods. Message semantics are determined by their structure and content.

The ACL ontology [25] is used to define message structure. The structure determines what to do with message content, which can be a query or metadata like those presented in section 2.3. The actions that can be taken by the broker are at last supported by the metadata store elements that allow metadata storage and retrieval.

3.1.1. Message Structure

Message structure is based on the Agent Communication Language. ACL [21] defines a set of communicative acts that establish message intentionality, i.e. its pragmatics. ACL also defines attributes that determine message characteristics. Some of these communicative acts are used in messages sent to the broker because they correspond to the user actions it manages:

- **Insert and Delete**: this action is captured by the **inform** communicative act when a chunk of metadata is "informed" to the broker. When a reference (URI) pointing to the metadata is communicated the **inform-ref** act is used. The **inform** can be used to assert affirmative or negative facts, i.e. unassert. The broker responds with a **inform** message to communicate insertion outcome.
- **Query**: this action corresponds to the **query-ref** act. It is a query by reference, where the reference is the pattern encoded by the query sentence. There are many

RDF query languages so the language attribute is used to tell the broker which one is used. The broker responds to the query with an **inform** message.

The message semantics defined by ACL are used by the broker to route them to the appropriate metadata store peer as detailed next. The appropriate store is determined by the broker, for instance by considering the message language attribute.

3.1.2. Metadata Storage

The different broker actions end up with an access to the metadata storage system. As it has been shown in the architecture, it has been separated from the broker in different independent modules. Communication between the broker and the selected metadata storage peer is also performed by means of ACL structured messages.

The message communicative act tells the store peer how it has to interpret it. The content of inform messages is inserted or deleted and query-ref messages content is interpreted as query sentences.

The store peer is supported by a RDF store that is in charge of really storing the metadata or retrieving the stored metadata corresponding to the pattern determined by the query sentence. The store peers make the broker and all the application independent from the particular RDF store used. Therefore, they show the same behaviour. They receive RDF metadata as input of Insert and Delete actions.

MetadataStore Peers are not only responsible for making the NewMARS system independent from the different metadata store particularities. Moreover, they are also responsible for converting metadata query results from the common table-like result sets to RDF metadata as it is detailed next.

3.1.3. Metadata Retrieval

As has been shown in the previous section, the Broker receives RDF metadata as input. This is a common behaviour of RDF stores so, in this case, little work has to be done.

On the other hand, as it has been stated during the application domain analysis, it is also very interesting to get RDF metadata as output from RDF stores so the whole information flow is done in RDF form. This has been justified as it facilitates the integration of metadata coming from different sources.

Moreover, if the web portal receiving the output from the NewMARS broker gets RDF metadata instead of table-like result sets, more information would be available in order to render this metadata to the user. In other words, the stored metadata semantics would not be lost in the query output and would arrive intact until the last information processing step.

In this case there is some work to do as producing RDF metadata as query output is a very uncommon behaviour of RDF metadata stores. Query sentences are augmented by the NewMARS Broker with a special construct "graph(*sentence*, *depth*)". When this construct is sent to a store peer, it indicates that the store peer has to construct one or more RDF graphs from the resources selected with the query sentence.

This is done by retrieving RDF triples from the selected resource to the maximum depth specified. However, blank nodes are not considered when computing this depth;

i.e. triples with blank node resources are always added if they are directly connected to selected resources or indirectly through a chain of blank resources.

For an example see Fig. 5. From a query that selects the resource "urn:newmars:30m-USAP", the graph construction algorithm is applied with depth equal to one. All the grey filled resources and literals and the solid line properties are retrieved. The Bag anonymous resource is ignored in order to compute depth so its members and its type are also retrieved. On the other hand, the metadata attached to the Video and Documentary types, the white filled resources and literals and the dotted line properties, are not retrieved as they are at a greater depth.

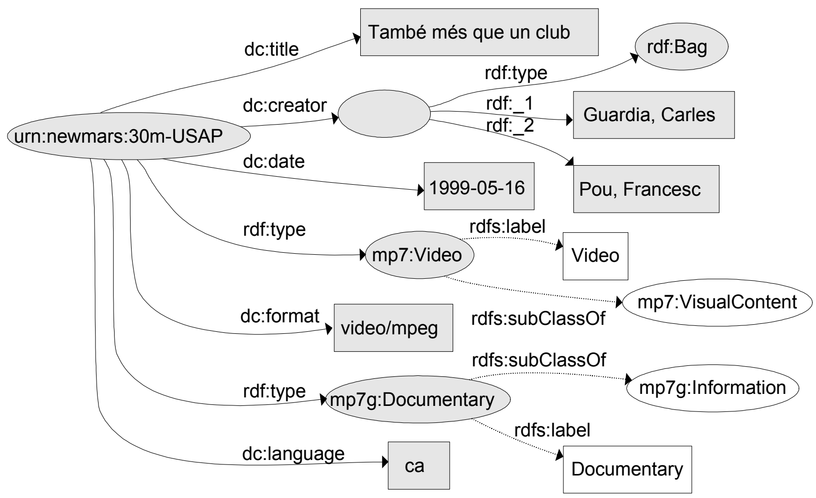

Fig. 5. Graph construction example for metadata retrieval

Once the query response graph or graphs have been constructed, they are serialised as RDF/XML and encapsulated in the response messages. They are structured as inform messages containing the response metadata.

As it has been shown in this and the previous section, store peers allow a great independence from the concrete RDF stores used. Currently two RDF stores have been integrated: RDF Suite [26] and Sesame [27].

3.2 Web Portal

The web portal has been developed as the user interface to the NewMARS functionality. The application has been developed based on the interchange of RDF messages with SOAP transport. Therefore, the portal must have a mechanism to encapsulate user interactions as RDF/ACL messages and send them to the broker by SOAP. Moreover, the responses to user interactions are made visible to the user by translating them to HTML. The web portal functionality is detailed in the next subsections.

3.2.1 RDFSOAPSender

This is the module responsible for the interaction between the portal and the broker. It is a servlet that receives user commands encoded as HTML form submissions. The form parameters are transformed into RDF triples, one for each parameter. All the triples have the same resource that identifies the current command. The properties are the parameter names and the resources their values.

The triples are serialised as RDFXML that is inserted into a new SOAP message in order to be sent, as shown in Table 2. The RDF content of the messages is built from the parameters received from the HTML forms through which the users interact with the system. Three basic forms can be identified: Query, Register/Offer and Retract.

Table 2. Example of SOAP envelope used to transport RDF/ACL messages

```
<SOAP-ENV:Envelope xmlns:... >
    <SOAP-ENV:Body>
        <rdf:RDF ... >...</rdf:RDF>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.2.2 Query Form

This form is composed by a set of fields relative to the attributes that finally will compose the RDF/ACL message that the RDFSOAPSender is going to generate. The available fields in the Query form are:

- **Sender**: the form web page URL or the identifier with which the user has identified himself in the web portal.
- **Receiver**: the broker URL where the SOAP message will be sent.
- **Reply-to**: the URL where the results will be sent in order to show them.
- **Content**: the query sentence.
- **Language**: the query language. Current RDF stores (RDF Suite and Sesame) use RQL. However, other possibilities can be easily incorporated.
- **Performative**: it indicates the message communicative act. For the query form it is fixed to the query-ref act.

Table 3 shows an example of RDF/ACL message built from a query form submission. It is an RQL query that retrieves metadata associated to offers that allow access to multimedia contents. The response is redirected to a web page that will format the output RDF metadata as HTML.

Table 3. Example RDF/ACL message built by RDFSOAPSender for a query form submission

```
<rdf:RDF  xmlns:acl="http://daml.umbc.edu/acldaml" ...>
<acl:query-ref>
    <acl:sender>http://dmag.upf.edu/newmars/search.html</acl:sender>
    <acl:receiver>http://dmag.upf.edu/newmars/broker</acl:receiver>
    <acl:language>RQL</acl:language>
    <acl:content parseType=Literal>
graph(select X,Y from {X;Offer}permission{;Access}.patient{Y;AudioVisual})
    </acl:content>
    <acl:reply-to>http://dmag.upf.edu/newmars/results.jsp</acl:reply-to>
</acl:query-ref>
```

3.2.3 Register/Offer Form

This form is used to tell the broker the IP descriptive, rights or e-commerce metadata to be inserted in the system. It is like the previous form. The only changes are performative, inform or inform-ref, and language that now is RDF/XML in order to reflex that the content is RDF metadata.

3.2.4 Metadata Web Rendering

The result web pages use XSL style sheets in order to transform the RDF metadata form response messages into HTML that can be shown by the web portal. There is a basic style sheet responsible for transforming each RDF description in the response metadata into an HTML table. Each row corresponds to one property directly associated to the description. The first column is the property id and the second column is the property value. If the value is another resource, a sub-table is recursively inserted and the whole table construction process is repeated.

This basic XSL style sheet is then combined with particular ones that complete HTML layout in order to particularise output to the special needs required. An example of complete HTML layout of a RDF encoded offer is shown in Fig. 6.

<table>
<tr><th colspan="2">http://dmag.upf.edu/newmars/offer19990611-103520</th></tr>
<tr><td>type</td><td>Offer</td></tr>
<tr><td>time</td><td>1999-06-11</td></tr>
<tr><td>offerer</td><td>NewMARSSeller@dmag.upf.edu:1099/JADE</td></tr>
<tr><td>patient</td><td>
<table>
<tr><td>type</td><td>PurchaseLicense</td></tr>
<tr><td>permission</td><td>
<table>
<tr><td>type</td><td>Access</td></tr>
<tr><td>timeFrom</td><td>1999-07-01</td></tr>
<tr><td>timeTo</td><td>2004-07-01</td></tr>
<tr><td>patient</td><td>urn:newmars:30m-USAP</td></tr>
<tr><td>place</td><td>http://www.tvcatalunya.com/online/30m-USAP</td></tr>
</table>
</td></tr>
<tr><td>obligation</td><td>
<table>
<tr><td>type</td><td>Compensation</td></tr>
<tr><td>input</td><td>
<table>
<tr><td>type</td><td>CurrencyMeasure</td></tr>
<tr><td>value</td><td>1</td></tr>
<tr><td>currencyUnit</td><td>http://www.daml...uk/ont/currency.daml#EUR</td></tr>
</table>
</td></tr>
<tr><td>payee</td><td>http://www.tvcatalunya.com</td></tr>
</table>
</td></tr>
<tr><td>licenser</td><td>http://www.tvcatalunya.com</td></tr>
</table>
</td></tr>
</table>

Fig. 6. HTML after the XSL transformation of the RDF-encoded Offer presented in Table 1

As has been shown in section 3.1.3, the metadata that is rendered is collected by building graphs from the selected resources to a given depth, commonly with depth one. In many cases this produces bunches of metadata with the relevant information for the posed query. However, sometimes it is necessary to get deeper in the graph and retrieve more metadata.

In order to facilitate metadata navigation, the XSL style sheet also produces HTML links for all the resource URLs. This links correspond to queries to the NewMARS broker for metadata about the clicked resource. Then, a pop-up window is opened showing the new metadata detail. The same XSL style sheet is applied to it so new HTML links are generated and they allow continuing the RDF metadata browsing experience through HTML. It can be tested on-line in the NewMARS web site [28].

3.3 Negotiation Support

Agents technology is used to perform negotiation. Negotiation is the last customer action. It is performed once the customer has located the desired content and the corresponding offer that is going to be negotiated. Offers can be directly accepted, rejected or negotiated.

We have chosen the JADE multiagent platform. In order to reason about facts coming from messages, JESS (Java Expert System Shell) [24] has been used because of its easy interoperability with JADE.

If the customer wants to negotiate the offer, he can choose a personal agent that will intermediate between the customer and the content provider agent. Customer and content provider agents are JADE agents controlled by the expert system. They negotiate the license offers.

The negotiation protocol is controlled by JESS and this allows a dynamic negotiation between the agents, making offers and counteroffers and processing licenses. There are two main phases in the negotiation that are only introduced in the next subsections. More details about the negotiation support are given in other publications from our group especially devoted to this issue [29,30].

3.3.1 First phase

Once the customer has chosen his representative agent, it is created and all the necessary data is loaded in the expert system. The metadata that models the negotiated offer and its context is loaded together with all the ontologies that define the concepts used by the metadata.

As has been already shown, all is expressed in RDF and OWL. In order to operate with JESS, all the metadata and ontologies are imported using OWLJessKB [31]. After that, the negotiation protocol and policies are also loaded. They are modeled as a set of rules in JESS format, i.e. CLIPS.

The protocol rules govern the timing of the different negotiation phases. On the other hand, the policy rules support the decision process of the agent. For instance, buy or sell only when a condition about price or duration is achieved.

This is an important feature because it allows us to flexibly determine important contract parameters as duration, prices and so on. Thus, we get a dynamic negotiation mechanism because negotiation policies can be easily changed and configured.

3.3.2 Second phase

In this phase the negotiation is finally carried out. The customer agent contacts the agent that is in charge of the offer negotiation. This is done using the information captured in the initial offer. There is the “offerer” property that identifies the corresponding agent using a JADE identifier, see Table 1.

The content provider agent that is responsible for negotiating the offer is the representative of the content provider that made the offer. It is ready to handle negotiations and pre-configured with the desired negotiation policy.

When it is contacted, it retrieves the negotiated offer from the NewMARS broker. It is loaded together with the received counteroffer and the required ontologies in the JESS engine that governs its behaviour.

The customer will then use the customer agent as the intermediary between him and the content provider agent. The customer agent can be more or less interactive, i.e. more or less autonomous. On the other hand, the content provider agent is totally autonomous and thus it takes decisions completely on its own, as specified in its negotiation policy.

The negotiation process goes on through the corresponding protocol as a series of offers and counteroffers. The final outcome can be an agreement if both parts agree on the offer conditions. These conditions will then constitute the license that is digitally signed by both parts. On the other hand, the negotiation process can fail if any part leaves the process.

4 Conclusions and Future Work

The main conclusion from the NewMARS development has been the great benefits that can be obtained from a knowledge oriented application. A high module independence based on the particular semantics has been achieved. This allows employing the same techniques for different domains by only adapting the conceptual framework, i.e. the ontologies that define the metadata structure.

For instance, in order to check NewMARS semantic capabilities, it has been also used with third party metadata. Concretely, it has been fed with RDF metadata from the MusicBrainz [32] project. This project has its own ontology for the music domain, i.e. album, track, artist,… The only effort necessary in order to make NewMARS manage resources annotated with MusicBrainz metadata has been to connect its ontology with the NewMARS ontological framework.

This has been easy thanks to the foundation of NewMARS ontological framework in IPROnto, a quite generic conceptualisation. Therefore, NewMARS can be easily configured to manage rights for any kind of intellectual property.

Our future plans include extending the NewMARS functionality in order to cope with a greater range of IP e-commerce scenarios. This can be reduced, thanks to the NewMARS knowledge oriented infrastructure, to ontologies modelling and negotiation policies definition. Moreover, NewMARS can deal with even more distant domains if the whole ontological support is properly adapted.

Finally, we are aligning NewMARS with MPEG-21 IPR management standardisation efforts. To achieve this, we are developing MPEG-21 compliant ontologies [33,34] that would allow NewMARS to manage standard IPR metadata.

Acknowledgements

NewMARS is part of the Agent Web project, partly supported by the Spanish administration TIC 2002-01336 (http://dmag.upf.edu/newmars). The final part of the work presented was developed within VISNET, a European Network of Excellence (http://www.visnet-noe.org), funded under the European Commission IST FP6 program.

References

1. Bormans, J. and Hill, K. (eds.): “MPEG-21 Overview”. ISO/IEC JTC1/SC29/WG11/N5231, 2002 http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm
2. ISO/IEC FDIS 21000-5, MPEG-21 Rights Expression Language (REL), ISO/IEC JTC 1/SC 29/WG 11/N5839, July 2003
3. ISO/IEC FDIS 21000-6, MPEG-21 Rights Data Dictionary (RDD), ISO/IEC JTC 1/SC 29/WG 11/N5842, July 2003
4. Bray, T.; Paoli, J.; Sperberg-McQueen, C. M. and Maler, E. (eds.): “Extensible Markup Language (XML) 1.0 (Second Edition)”. W3C Recommendation 6 October 2000 http://www.w3.org/TR/REC-xml
5. Iannella, R.: Open Digital Rights Language (ODRL) Version 1.1. W3C Note, 2002 http://www.w3.org/TR/odrl
6. Berners-Lee, T.; Hendler J. and Lassila O.: “The Semantic Web”. Scientific American, May 2001 http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21
7. Hendler, J. “Agents on the Semantic Web”. IEEE Intelligent Systems, Vol. 16, No. 2, March-April 2001 http://www.ai.mit.edu/people/jimmylin/papers/Hendler01.pdf
8. Intellectual Property Rights ONTOlogy, http://dmag.upf.edu/ontologies/ipronto
9. Delgado, J.; Gallego, I.; Llorente, S. and García, R.: “Regulatory Ontologies: An Intellectual Property Rights approach”. Workshop on Regulatory ontologies and the modeling of complaint regulations, WORM Core. LNCS, Vol. 2889, pp 621-634, 2003
10. Delgado, J.; Gallego, I.; Llorente, S. and García, R.: “IPROnto: An Ontology for Digital Rights Management”. 16th Conference on Legal Knowledge and Information Systems, JURIX. Frontiers in Artificial Intelligence and Applications, Vol. 106, IOS Press, 2003
11. “The IMPRIMATUR Business Model, Version 2.1”. IMPRIMATUR Project (Esprit 20676), WP4, 1999. http://www.imprimatur.net/IMP_FTP/BMv2.pdf
12. Gallego, I.; Delgado, J. and Acebron, J.J.: “Distributed models for brokerage on electronic commerce”. In Trends in distributed systems for electronic commerce, 1998
13. Delgado, J., Gallego, I., Polo, J. "Electronic commerce of multimedia services", in MultiMedia Modeling, World Scientific Publishing, pp. 97-110, 1999
14. García, R.; Delgado, J.: “Brokerage of Intellectual Property Rights in the Semantic Web”. 1st Semantic Web Working Symposium (SWWS-1), Stanford, CA, 2001
15. Martinez, J. “Overview of the MPEG-7 Standard”. MPEG Document ISO/IEC JTC1/SC29/WG11 N4509, 2001 http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm
16. Hunter, J. “An RDF Schema/DAML+OIL Representation of MPEG-7 Semantics”. MPEG Document ISO/IEC JTC1/SC29/WG11 W7807, 2001
17. ISO/IEC 15938-5 FDIS Information Technology. “Multimedia Content Description Interface - Part 5: Multimedia Description Schemes”. ISO/IEC JTC1/SC29/WG11 Document W4242, 2001
18. Hunter, J. “An Application Profile which combines Dublin Core and MPEG-7 Metadata Terms for Simple Video Description”. Harmony Project Draft, 2002 http://metadata.net/harmony/video_appln_profile.html
19. Mitra, N. (ed.) “SOAP Version 1.2 Part 0: Primer”. W3C Working Draft, W3C XML Protocol Working Group, 2002 http://www.w3.org/TR/soap12-part

20. Lassila, O. and Swick, R.R. (eds.): “Resource Description Framework (RDF)”. Model and Syntax Specification. W3C Recommendation 22 February 1999 http://www.w3.org/TR/REC-rdf-syntax
21. “FIPA ACL Message Structure Specification”. FIPA Agent Communication Language Specifications, Document Id. XC00061, 2002 http://www.fipa.org/specs/fipa00061
22. Dean, M. and Schreiber, G. (eds.): “OWL Web Ontology Language Reference”. W3C Proposed Recommendation, Web Ontology Working Group, 2003 http://www.w3.org/TR/owl-ref
23. Bellifemine, F. ; Caire, G. ; Poggi, A.; Rimassa, G. : “JADE - A White Paper”. Telecom Italia Lab, EXP Online, Vol. 3. n. 3, 2003 http://exp.telecomitalialab.com/upload/issues/v3n3.pdf
24. Friedman-Hill, E.: “Jess in Action: Rule-Based Systems in Java”. Manning Publications Co., 2003
25. ACL DAML Ontology, http://www.cs.umbc.edu/~yzou1/daml/acl.daml
26. ICS-FORTH RDF Suite, http://www.ics.forth.gr/proj/isst/RDF
27. Sesame, http://www.openrdf.org
28. NewMARS Web Site, http://dmag.upf.edu/newmars
29. Gil, R.; García, R. and Delgado, J.: “Delivery context negotiated by mobile agents using CC/PP”. Int. Conference on Mobile Agents for Telecommunication Applications, MATA’03. LNCS, Vol. 2881, pp 99-110. Springer-Verlarg, 2003
30. Delgado, J.; Gallego, I.; García, R.; Gil, R.: “An architecture for negotiation with mobile agents”. Int. Conference on Mobile Agents for Telecommunication Applications, MATA’02. LNCS, Vol. 2521, pp 21-31. Springer-Verlarg, 2002
31. Kopena, J. and Regli, W.: “DAMLJessKB: A Tool for Reasoning with the Semantic Web”. IEEE Intelligent Systems, Vol. 18, No. 3, pp. 74-77, 2003
32. MusicBrainz, http://www.musicbrainz.org
33. García, R.; Delgado, J.; Rodríguez, E.; Llorente, S. and Gallego, I.: “RDDOnto, Rights Data Dictionary Ontology Version 2”. ISO/IECJTC1/SC29/WG11/M10423, 2003
34. Delgado, J.; Gallego, I.; García, R.: “Use of Semantic Tools for a Digital Rights Dictionary”. Accepted in EC-WEB’04 conference. To be published in LNCS, 2004